

10. TP02 : Années bissextiles : booléens, opérateurs and et or.

Les années comptent 365 jours dont 28 en février, sauf les années bissextiles à 366 jours dont 29 en février.

Les années bissextiles sont les années :

- Dont le millésime est divisible par 4, mais pas par 100
- Et les années dont le millésime est divisible par 400

1. Parmi les années suivantes, quelles sont les années bissextiles ?

1900, 1983, 2000, 2016, 2018, 2020, 2100.

2. (a) Proposer un organigramme correspondant à la situation.

(b) Transcrire cet organigramme sous forme d'algorithme en "langage naturel".

(c) Ouvrir EduPython, et y coder le programme correspondant.

3. Booléens, tests d'égalité, connecteurs logiques and et or.

Un **booléen** est un type de variable qui ne peut prendre que deux valeurs : True (vrai) ou False (faux). Lorsque l'on évalue une expression booléenne, on obtient un booléen.

Par exemple, l'égalité " $x = 4$ " est une expression booléenne : elle est vraie si la variable x vaut 4, et fausse sinon.

De même, l'expression " $3x + 1 \neq 2$ " est une expression booléenne.

En Python, le test d'égalité se note " $=$ ", et le test d'inégalité se note " $!=$ ". Le symbole $=$ sert à **affecter une valeur à une variable**.

Ainsi, les deux expressions booléennes ci-dessus se notent respectivement " $x == 4$ " et " $3 * x + 1 != 2$ ".

(a) Écrire cinq exemples d'expressions booléennes.

(b) En Python, le **connecteur logique** "et" se note "**and**" et "ou" se note "**or**" (comme en anglais).

Attention, en Python comme en mathématiques, "ou" signifie "et/ou".

Utiliser les tests " $==$ ", " $!=$ " et les connecteurs logiques "**and**" et "**or**" pour écrire trois nouveaux exemples d'expressions booléennes.

(c) En Python, l'**opérateur modulo** se note "**%**". Il s'agit du reste d'une division euclidienne. Par exemple, $9 \% 4 = 1$ car le reste de la division euclidienne de 9 par 4 est 1.

Utiliser l'opérateur "**%**" pour traduire la condition : "le nombre n est divisible par 400".

4. Mise en pratique :

(a) A l'aide des expressions booléennes : **annee % 4 == 0** **annee % 100 != 0** **annee % 400 == 0**
et des connecteurs logiques **and** et **or**

former une unique expression booléenne en Python qui prend la valeur **True** si la variable *annee* contient une année bissextile, et **False** si la variable *annee* contient une année non bissextile.

(b) Modifier le petit programme de la question 2c pour qu'il utilise cette expression booléenne.

Les élèves qui codent déjà en Python pourront proposer une fonction "bissex" qui renvoie True ou False selon que l'année est bissextile ou pas, et qui prendra en paramètre la variable "annee".

Ils pourront ensuite appeler cette fonction dans une autre fonction "nb_bissex" qui compte le nombre d'années bissextiles dans les n années qui suivent une année donnée. Cette fonction prendra en paramètre l'année de départ (variable "debut") et le nombre n d'années à considérer.

On obtiendra un code de ce type (à compléter avec le booléen défini à la question 4 :

```
def bissex(annee):  
    return ...  
  
def nb_bissex(debut, n):  
    compt=0  
    for k in range(n):  
        annee=debut+k  
        if bissex(annee):  
            compt=compt+1  
    return compt
```

L'appel **nb_bissex(2019, 10000)** donne comme sortie 2425. Interpréter ce résultat.

L'année durant en réalité 365,2422 jours, combien de jours d'avance notre calendrier prend-il en 10 000 ans ?